

FHWS

iOS Programmierung (mit Swift)

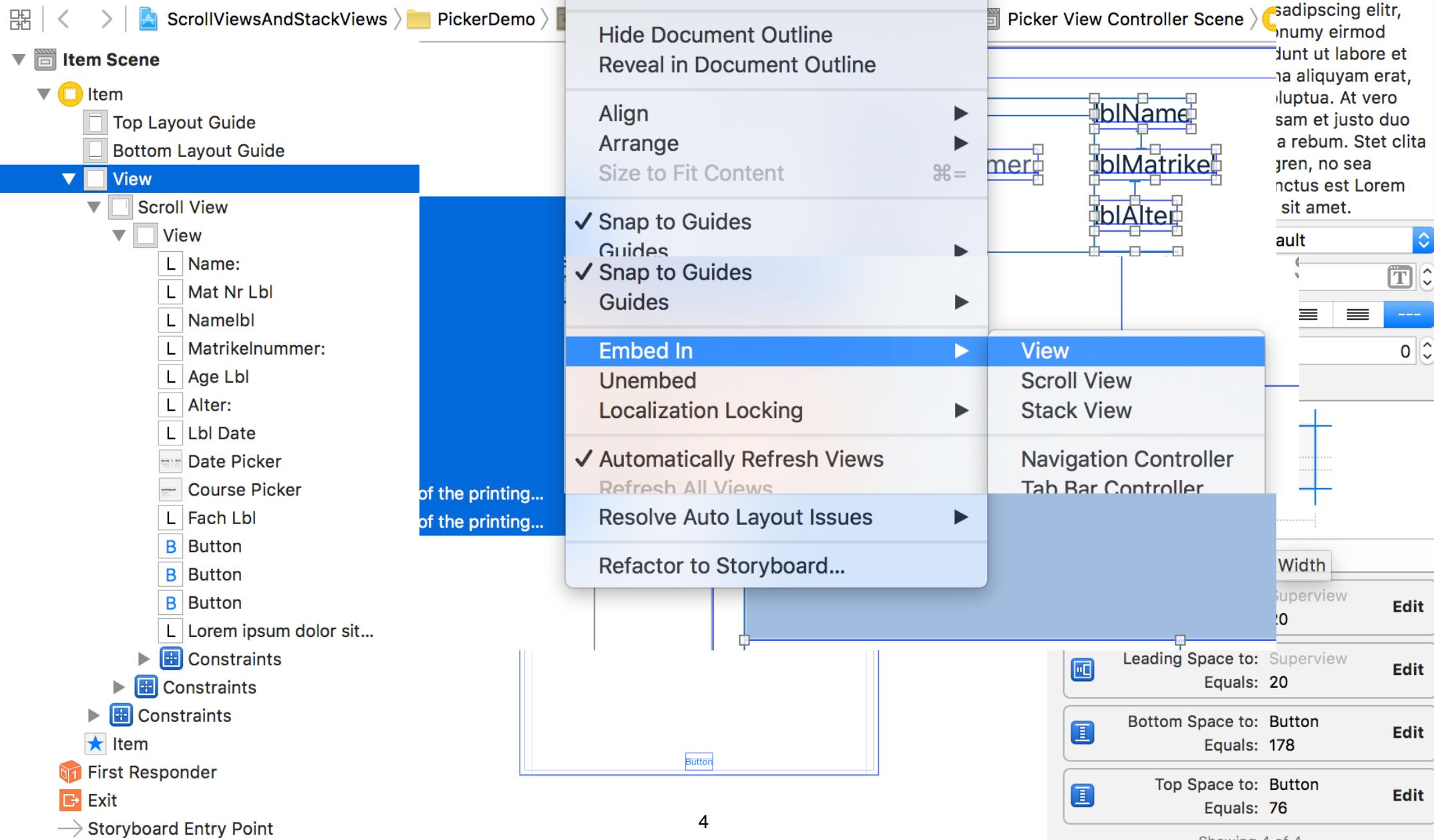
Peter Braun, Florian Bachmann & Andreas Wittmann
[@pe_braun](https://twitter.com/pe_braun) [@florianbachmann](https://twitter.com/florianbachmann) [@anwittmann](https://twitter.com/anwittmann)

Deutsche Telekom AG
FHWS - Hochschule für angewandte Wissenschaften Würzburg-Schweinfurt
#FHWSSwift

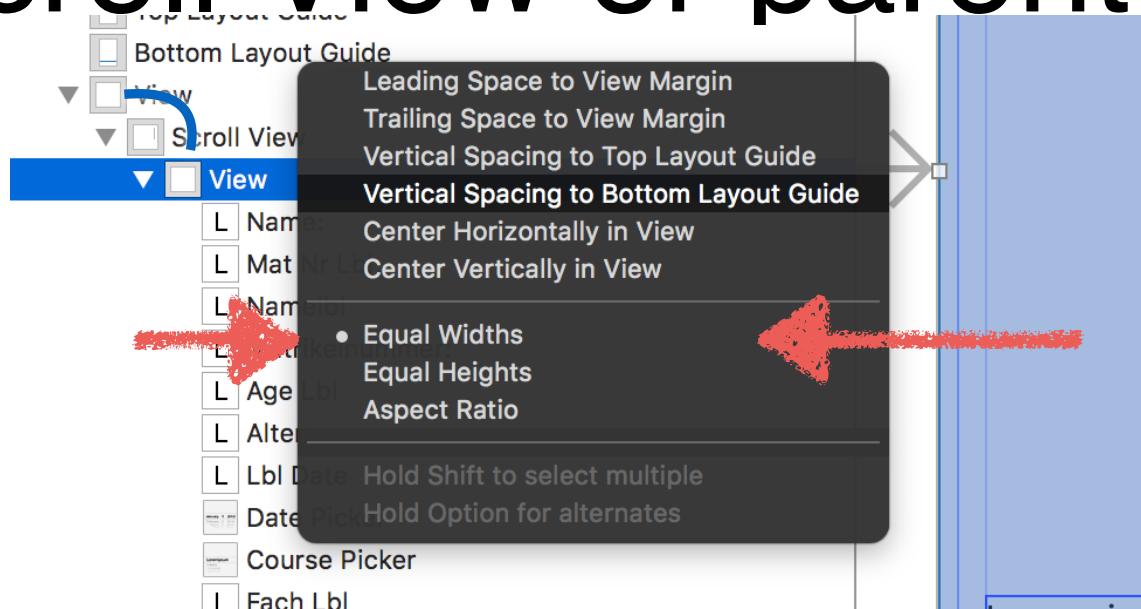
Agenda

1. **Introduction** – Organisatorisches
2. **First iOS-Project** – Hello World, **First iOS-Project** – Still Hello World (now with Code 😊)
3. **Swift**, Wait!, What about Objective-C?, Why Swift?
4. **A (not so) Quick Tour**
5. **Documentation**
6. **The basics** – iOS Architecture & more
7. **User Interfaces** – View Controller, Auto Layout & Size Classes
8. **Storyboard & Segues**
9. **Tables & NavigationController**
10. **TabBarController**
11. **Notifications**
12. **PickerViews**
13. **Touches, Gestures, 3D Touch, Peek & Pop**
14. **ScrollView & StackViews**
15. **Networking** – JSON & Dependency Managers
16. **WebKit**
17. **Maps**
18. **Storage & Data persistency** – NSUserDefaults, NSKeyedArchiver & Core Data
19. **ObjC**

ScrollViews
because there is so
much to show



Content view needs equal width to Scroll View or parent view



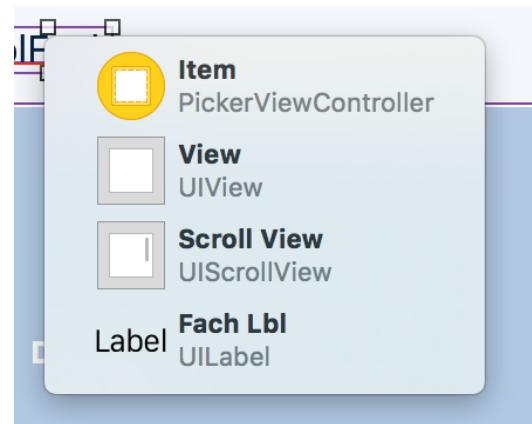
Mysteries of AutoLayout

X, Y, Width & Height
must be defined



Text has intrinsic content size

Ctrl + Shift + Left (or Shift + Right Click)



StackViews

yeah yeah yeah - the
LinearLayout of iOS

••••• ⌘

9:41 AM

100% 🔋

Back



GarageBand 4+

Apple >

Offers In-App Purchases

Essentials

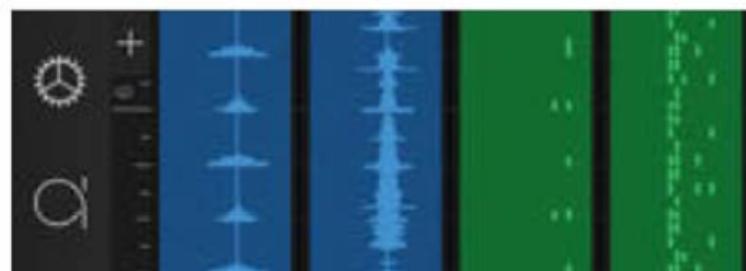
★★★★★ (1,085)

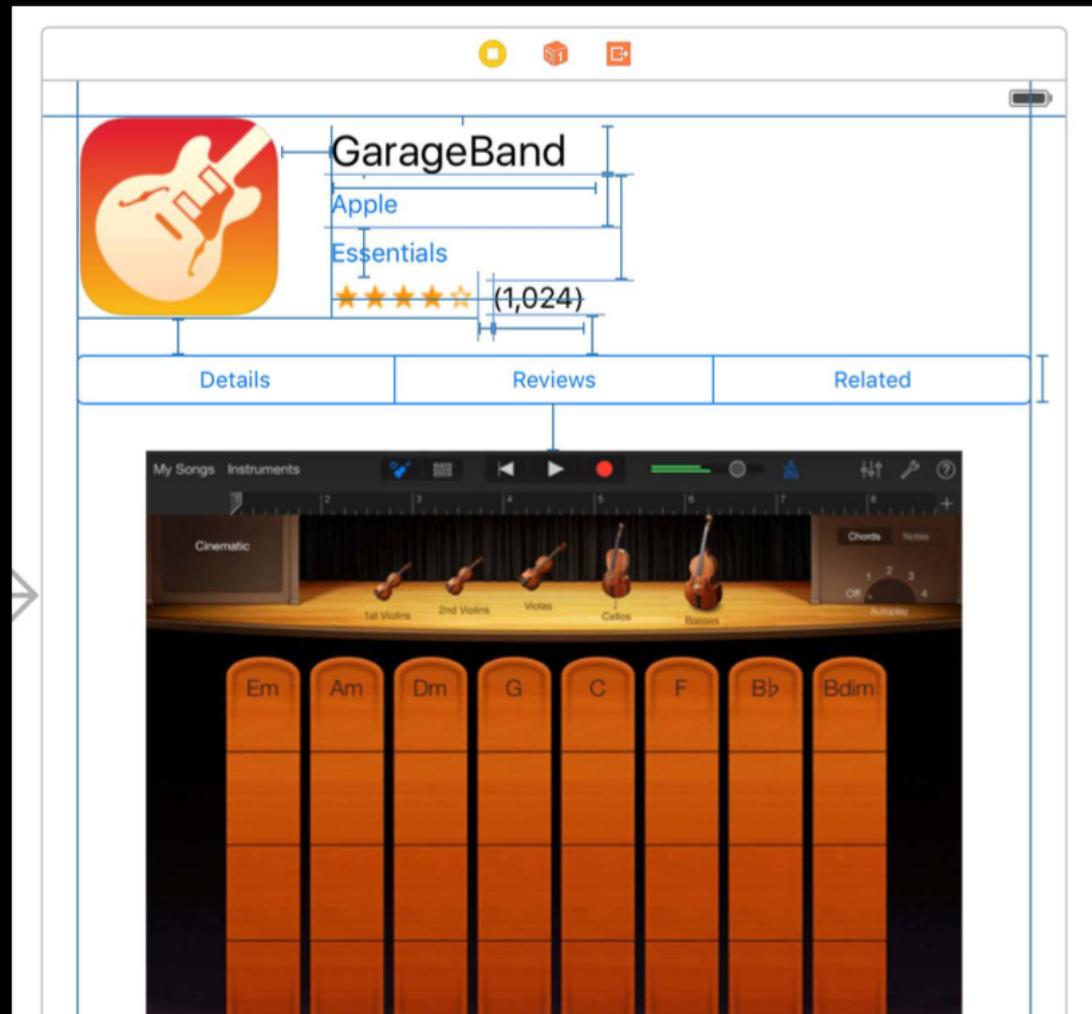
+ OPEN

Details

Reviews

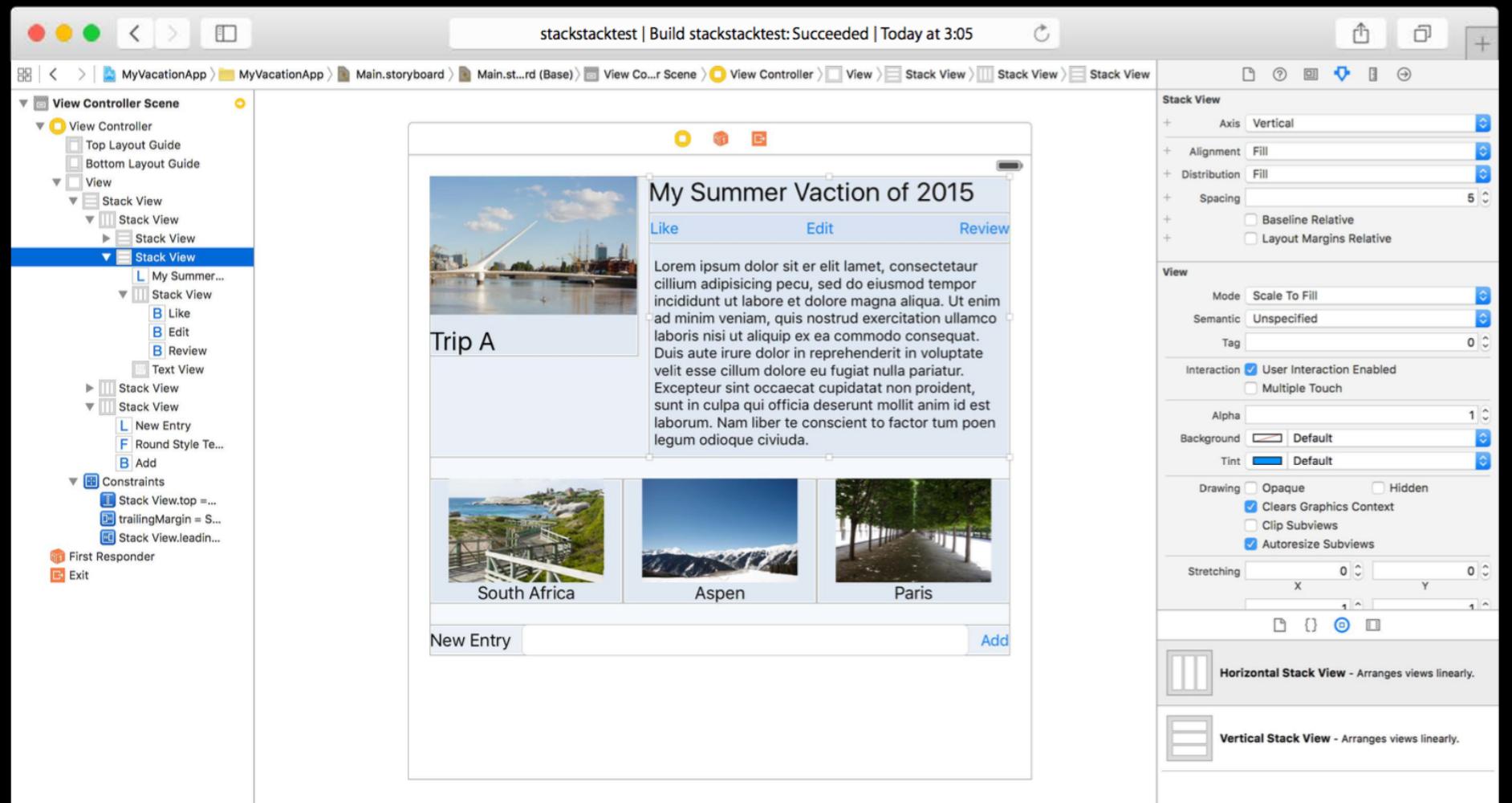
Related





Stack View

UIStackView (iOS 9) and NSScrollView (OS X 10.9)



Stack View

NEW

Built with Auto Layout

Manages constraints

Horizontal or vertical



Stack View

Alignment



Stack View Alignment



Stack View Distribution



Demo

Stack View in Interface Builder

```
// iOS 9  
class UIStackView {  
    var axis:UILayoutConstraintAxis  
    var distribution:UIStackViewDistribution  
    var alignment:UIStackViewAlignment  
    var spacing:CGFloat  
    func addArrangedSubview(view: UIView)  
    var arrangedSubviews: [UIView]
```

Stack View in Interface Builder

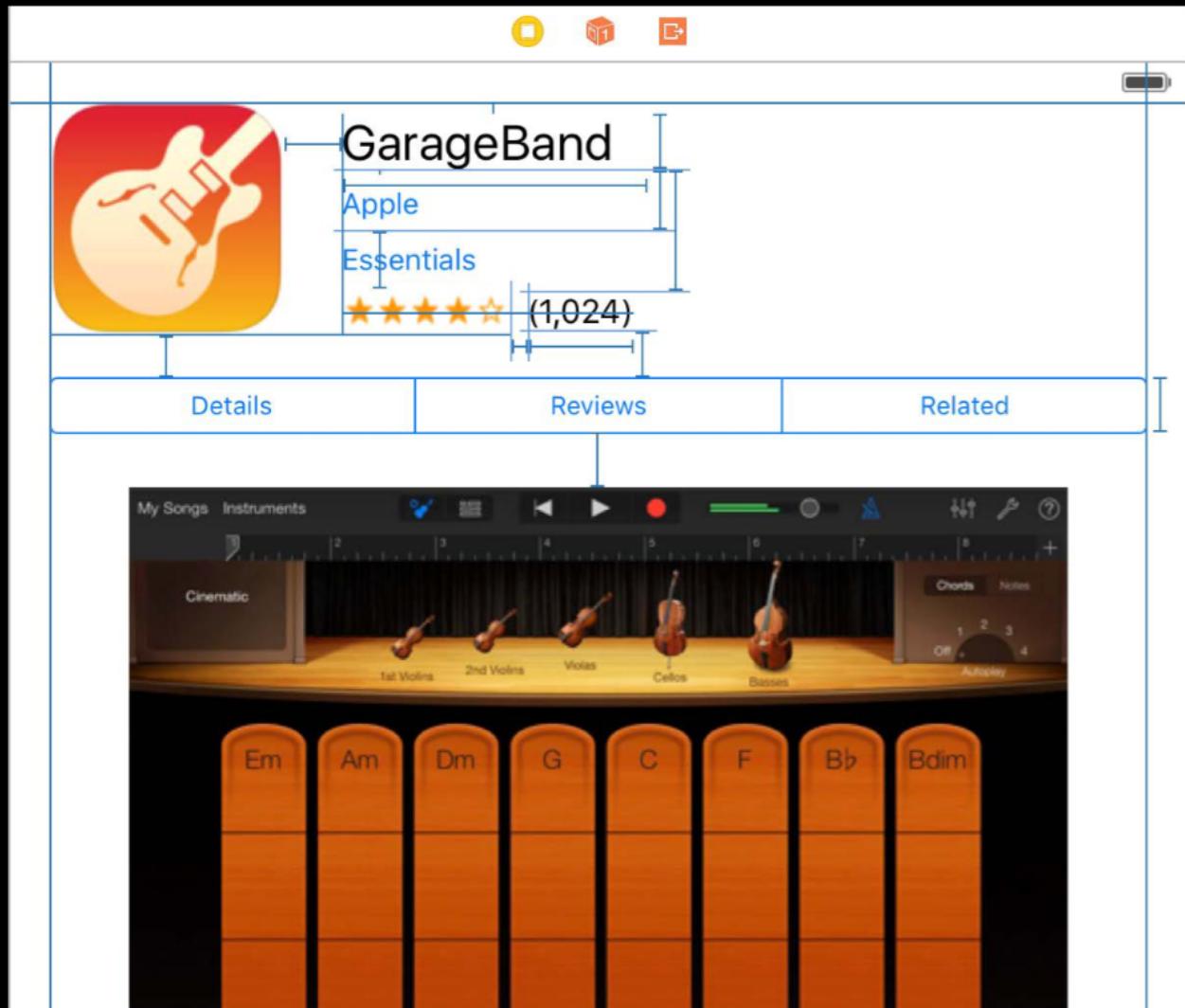
Easy to build

Easy to maintain

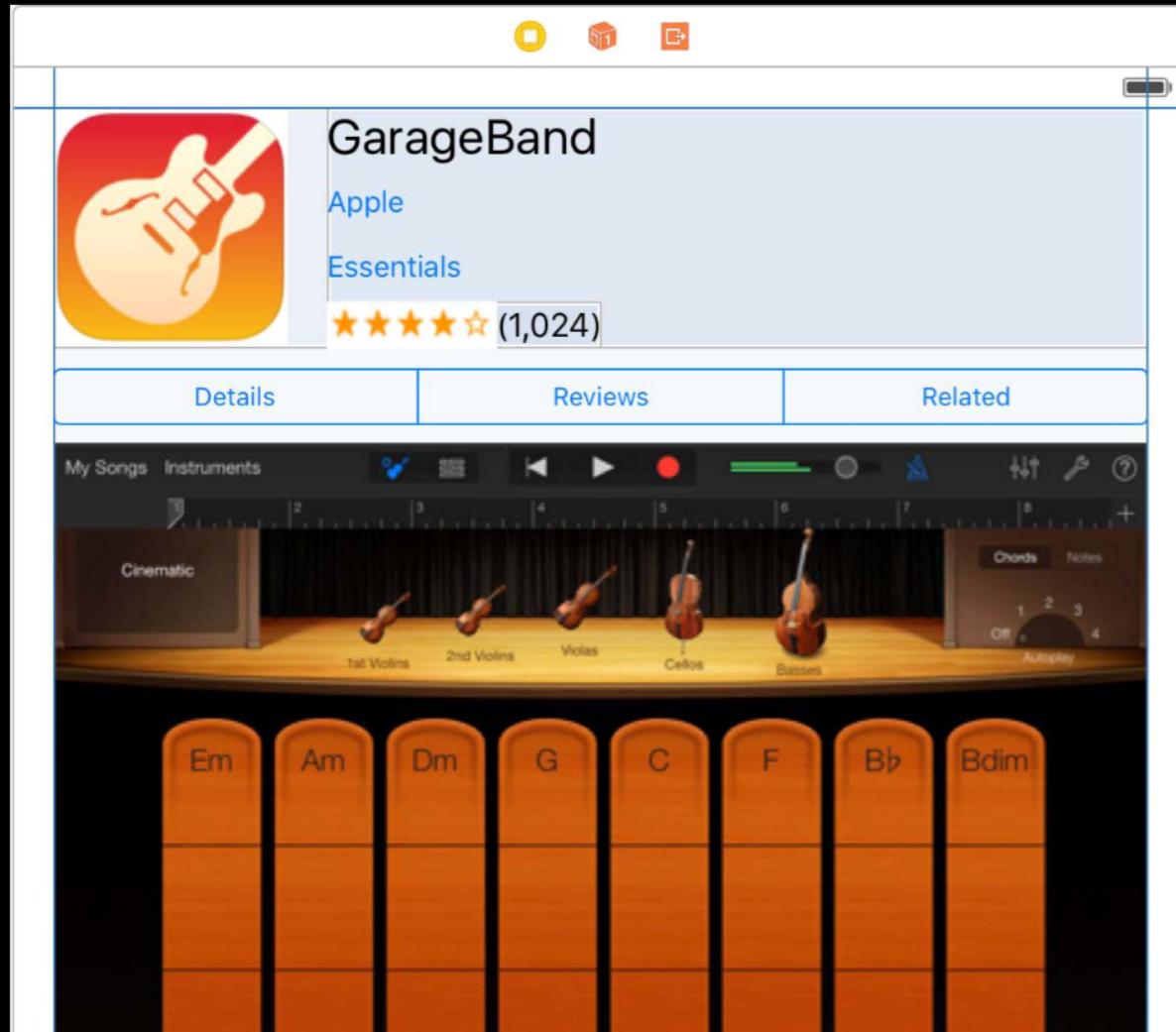
Composable Stack Views

Lightweight

Before, many constraints



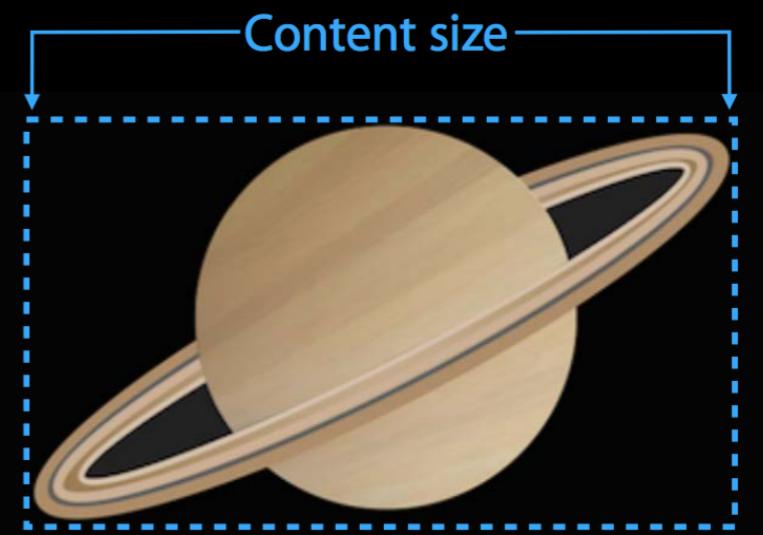
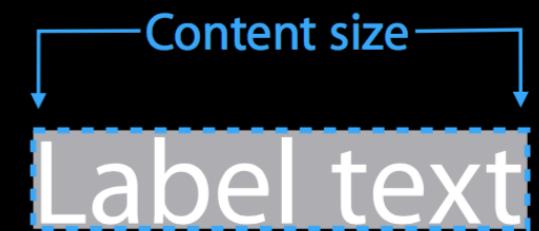
After, four constraints with Stack View



Start with Stack View,
use constraints as needed

View Size

Intrinsic content size



View Size

Intrinsic content size

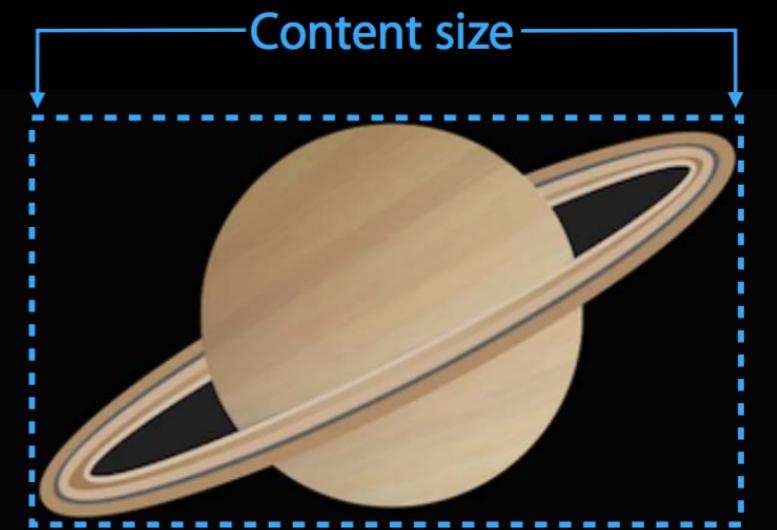
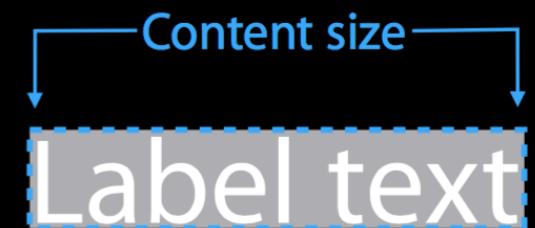
Certain views have an `intrinsicContentSize`

- For instance—labels and image views

Size derived from non-constraint internals

System makes the size constraints

Layout size is not guaranteed



Self-Sizing Table View Cells

Self-sizing needs size from constraints

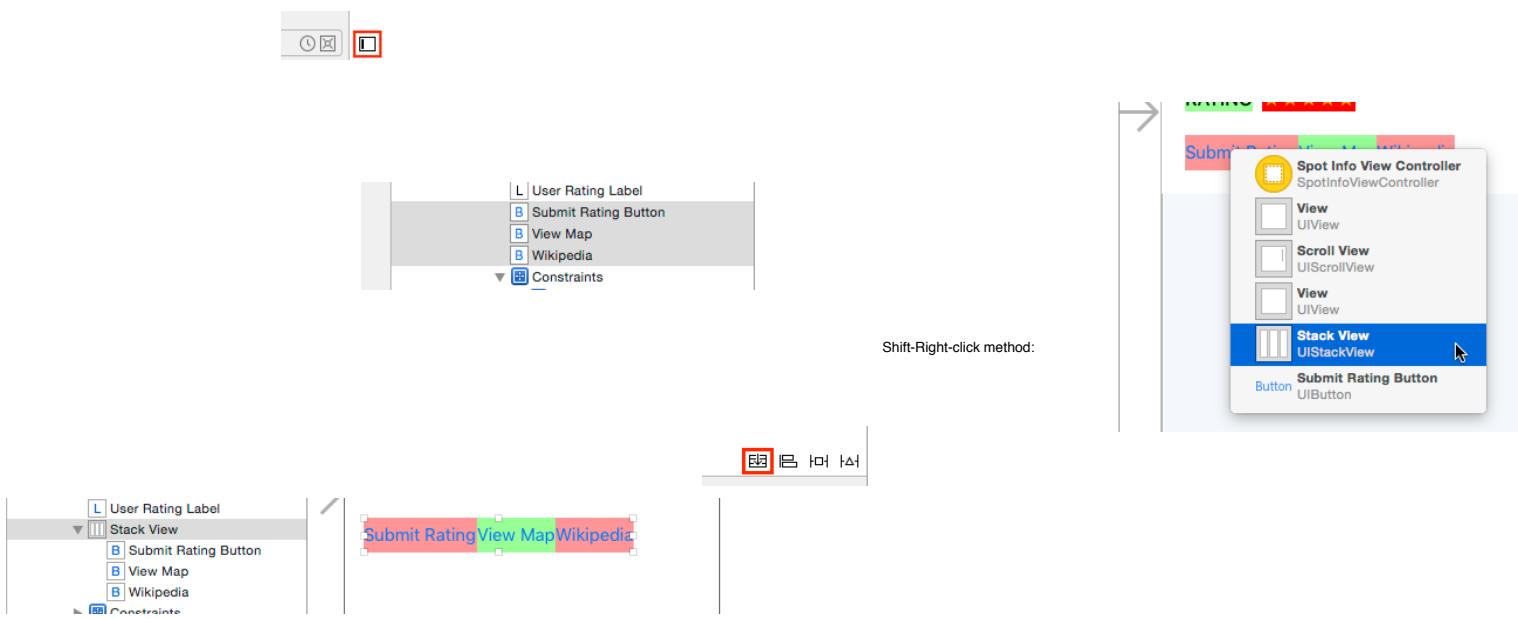
Width is defined with table view cells

Constraints must determine height

- Take advantage of proportions



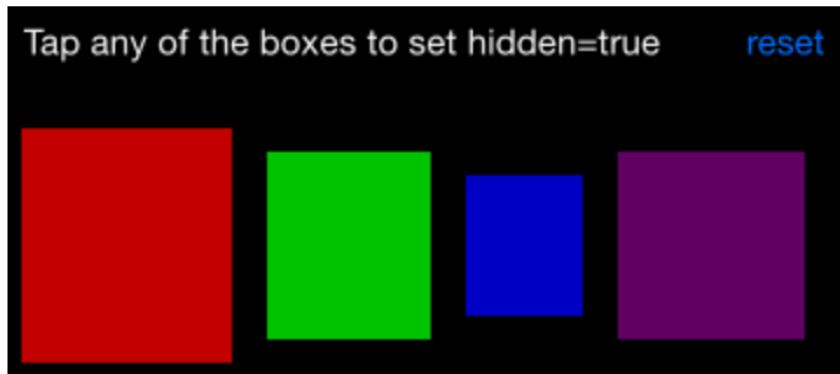
Isn't this view incredible? I wanted to stay until the stars came out, but Kris forgot that it gets cold here and didn't bring a sweatshirt, so we had to leave before it got too dark. :P



Demo

Self-sizing Table View Cells

property of a UIView, which normally is not an animatable property.



So all that's needed is animate the hidden property, like this:

```
1 | UIView.animateWithDuration(0.5, animations: {  
2 |     someSubview.hidden = true  
3 | })
```

Constraint Priorities

Priorities go from 1–1000

Content Priorities

How a view handles its content

By default, these are not set as required

- Do not set as required
- Can cause unsatisfiable constraints

Equal priorities can cause ambiguity

Types

- Content hugging

Content hugging priorities



251

Content Priorities

How a view handles its content

By default, these are not set as required

- Do not set as required
- Can cause unsatisfiable constraints

Equal priorities can cause ambiguity

Types

- Content hugging
- Compression resistance

Compression resistance priorities



750

750

Content Priorities

How a view handles its content

By default, these are not set as required

- Do not set as required
- Can cause unsatisfiable constraints

Equal priorities can cause ambiguity

Types

- Content hugging
- Compression resistance

Compression resistance priorities



czerwony

Priorities

Can help keep constraints from unsatisfiability

- But look out for competing priorities!

Results are more consistent

Use content priorities to get to the right layout

- Hugging priorities hug content
- Compression resistance resists squishing

Summary

Stack Views help build easily maintainable layouts

Use activate and deactivate for constraints

Determine size through constraints

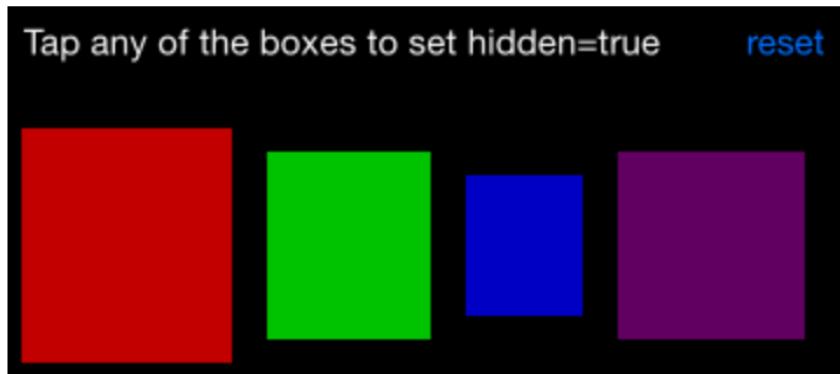
- Override `intrinsicContentSize` judiciously

Use priorities to properly solve your layout

Alignment goes beyond top, bottom, and center

- Keep localization in mind

property of a UIView, which normally is not an animatable property.



So all that's needed is animate the hidden property, like this:

```
1 | UIView.animateWithDuration(0.5, animations: {  
2 |     someSubview.hidden = true  
3 | })
```